

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Tabel 2.1
Penelitian Terdahulu

No	Judul, Penulis, Tahun	Data Masukan / Komponen	Model / Teknologi	Capaian / Hasil Evaluasi	Research Gap
1	Peran Smart Farming, IoT, dan Pertanian Presisi dalam Mewujudkan Ketahanan Pangan Berkelanjutan — P. S. Ningsi (2025) [14]	Data dan hasil kajian dari berbagai penerapan smart farming, IoT, otomatisasi, pertanian presisi, serta kondisi tanah dan iklim	Smart Farming, IoT, dan Precision Agriculture	Kajian menunjukkan bahwa smart farming dan IoT mendukung pemantauan kondisi tanah dan iklim secara real-time, sedangkan otomatisasi dan pertanian presisi dapat membantu mengoptimalkan penggunaan input seperti air, pupuk, dan pestisida. (Multidisiplin Ilmu Pasadena Journal)	Bersifat kajian/review dan belum menghasilkan sistem Smart Sprayer yang mengintegrasikan monitoring lingkungan, aktuator penyemprotan, dashboard, dan notifikasi pada tanaman bawang merah.
2	Pemodelan Sistem Monitoring Sensor Curah Hujan Menggunakan	Sensor curah hujan, Raspberry Pi, database InfluxDB	Grafana, InfluxDB, cloud monitoring	Sistem mengolah data sensor curah hujan melalui Raspberry Pi, menyimpan data pada InfluxDB, kemudian	Fokus pada monitoring dan visualisasi data curah hujan, belum diarahkan pada

	an Grafana — T. K. Widagdo, I. T. Bayu, dan Y. A. Susetyo (2018/2019) [15]	, cloud VPS		memvisualisasikannya menggunakan Grafana dalam bentuk grafik sehingga data lebih mudah dianalisis. (Ejournal UKSW)	pengambilan keputusan penyemprotan otomatis dan belum terintegrasi dengan sistem pengendalian sprayer pertanian.
3	Implementasi Smart Tree Tagging Untuk Monitoring dan Pemeliharaan Pohon Alpukat Dalam Program Pemberdayaan Masyarakat — I. Kumara, G. Dana, I. Sinarta, dan D. Enda (2025) [16]	Data identitas dan kondisi pohon, Barcode, NFC, dashboard Smart Farm	Smart Tree Tagging, Barcode, NFC, Dashboard, AI	Sistem membantu digitalisasi pencatatan dan pemantauan tanaman alpukat yang sebelumnya banyak dilakukan secara manual. Teknologi Smart Tree Tagging terintegrasi dengan Dashboard Smart Farm Warmadewa dan sistem kecerdasan buatan. (Journal of Community Development)	Berorientasi pada identifikasi, pencatatan, dan pemeliharaan pohon, belum mengintegrasikan sensor lingkungan dan aktuator penyemprotan otomatis untuk pengendalian hama bawang merah.
4	Implementasi Teknologi Irigasi Tetes pada Tanaman Jagung Menggunakan Sensor Soil Moisture dan Mikrokontroler ESP32 — M. J. Saputra dan	Sensor soil moisture, ESP32, pompa air, tanaman jagung	Otomatisasi berbasis threshold / rule-based	Sistem mendeteksi kelembapan tanah secara real-time dan mengaktifkan pompa secara otomatis ketika nilai kelembapan berada di bawah ambang batas. (Irpi Journal)	Fokus pada pengairan tanaman jagung, bukan pengendalian hama atau penyemprotan pestisida. Parameter yang digunakan juga terbatas pada kelembapan tanah dan

	R. R. Suryono (2024) [17]				belum mengintegrasikan sensor cuaca, dashboard monitoring, serta notifikasi peringatan.
5	Implementasi Algoritma RFID Pada Sistem Portal Parkir Otomatis Berbasis IoT Menggunakan Kartu Tanda Mahasiswa Di Universitas Dirgantara Marsekal Suryadarma — I. Santoso dan R. M. Sultoni (2025) [18]	RFID reader, KTM, ESP32, sensor deteksi kendaraan, portal otomatis, database web	RFID, ESP32, IoT, database web, otomasi aktuator	Sistem mampu melakukan validasi kartu, mengendalikan portal secara otomatis, dan mencatat data secara real-time. Pengujian menunjukkan validasi dan pengoperasian portal berjalan dengan akurasi tinggi serta respons yang cepat. (Universitas Suryadarma Journal)	Domain penelitian berada pada sistem parkir, sehingga belum menerapkan konsep otomasi aktuator pada monitoring dan penyemprotan pertanian serta belum mengolah parameter lingkungan tanaman.
6	Penelitian ini: Perancangan Sistem Monitoring dan Penyemprotan Otomatis Berbasis Website untuk Pengendalian Hama Kutu pada Bawang	Sensor suhu, kelembapan udara, kelembapan tanah, curah hujan, ESP32, pompa sprayer, database MySQL	IoT, threshold/rule-based, Laravel, website, WhatsApp API	Sistem dirancang untuk menampilkan kondisi lingkungan secara real-time, menyediakan kontrol sprayer manual dan otomatis, menyimpan riwayat sensor dan penyemprotan, serta	Mengintegrasikan monitoring lingkungan, pengendalian sprayer, website, penyimpanan data, dan notifikasi dalam satu sistem yang diarahkan untuk pengendalian

	Merah di Brebes — Sabrina Amania Astrid Susanto (2026)			mengirimkan notifikasi peringatan melalui WhatsApp API.	hama kutu pada tanaman bawang merah.
--	--	--	--	---	--------------------------------------

Sumber : dikelola peneliti, 2026

2.1.1 Deskripsi Penelitian Terdahulu

Penerapan *Internet of Things (IoT)* menawarkan mekanisme otomatisasi yang terbukti meningkatkan efisiensi dan presisi tata kelola pertanian[14]. Beberapa kajian terdahulu memanfaatkan Grafana, sebuah platform visualisasi open-source dari Grafana Labs, untuk menampilkan data pemantauan dalam format grafik secara *real-time*. Studi tersebut mengonfirmasi keandalan sensor *IoT* dalam mengumpulkan parameter lingkungan seperti kelembapan dan suhu. Kendati demikian, sistem yang dikembangkan masih bersifat pasif karena hanya terbatas pada fungsi monitoring tanpa adanya kendali aksi otomatis terhadap aktuator di lapangan[15].

Oleh karena itu, penelitian ini berupaya menjembatani kesenjangan antara kompleksitas teknologi *IoT* dengan kebutuhan praktis petani di lapangan. Selaras dengan hal tersebut, Kumara (2025) merancang platform monitoring tanaman berbasis *web* yang mampu menyajikan riwayat data sensor dalam bentuk tabel dan grafik interaktif. Namun, keterbatasan utama sistem ini adalah belum tersedianya modul kontrol dua arah untuk menggerakkan perangkat pompa secara remote. Petani masih dituntut melakukan intervensi manual setelah membaca visualisasi data, yang mana kelemahan ini diatasi dalam penelitian sekarang melalui implementasi otomatisasi sistem secara menyeluruh [16].

Saputra (2024) mengemukakan bahwa implementasi irigasi otomatis berbasis *IoT* menggunakan ESP32 and *rule-based* system efektif mengoptimalkan efisiensi sumber daya air, walaupun antarmuka monitoring interaktifnya masih perlu disempurnakan. Digitalisasi serta otomatisasi menjadi pilar penting bagi modernisasi pertanian demi mewujudkan keberlanjutan (*sustainability*) dan ketahanan pangan. Sebagai komparasi global, pemanfaatan *IoT* berbasis *cloud* dan robotika di Jepang berhasil menjaga stabilitas produksi pertanian di tengah krisis tenaga kerja. Adopsi teknologi ini terbukti meningkatkan produktivitas sebesar 20-30% sekaligus meminimalkan pemborosan air dan pestisida berkat sistem presisi. Dengan demikian, integrasi sensor dan otomasi merupakan solusi krusial bagi pengembangan *smart farming* yang ramah lingkungan[17].

Mengenai sistem notifikasi, Santoso (2025) menegaskan bahwa pembacaan sensor memberikan data *real-time* yang objektif untuk mendukung pengambilan keputusan. Sayangnya, efektivitas sistem kerap berkurang apabila ketiadaan fitur peringatan langsung membuat petani terlambat mengetahui anomali di lapangan. Mengingat *WhatsApp* merupakan aplikasi perpesanan yang mendominasi sekitar 83% pengguna internet di Indonesia, integrasi *WhatsApp Gateway* ke dalam platform monitoring berbasis *web* merupakan langkah strategis. Pemanfaatan *WhatsApp API* memungkinkan transmisi pesan peringatan dini (*early warning*) berjalan otomatis dan instan, memastikan langkah preventif tetap berjalan sekalipun pengguna sedang tidak memantau *dashboard website*[18].

Sistem yang dikembangkan dalam tugas akhir ini berupa *web* terintegrasi *IoT* yang dirancang untuk monitoring dan kontrol lahan secara komprehensif.

Aplikasi menyajikan data sensor kelembapan tanah, suhu udara, dan curah hujan dalam bentuk grafik *real-time* dan tabel historis guna mempermudah analisis tren tanah. Lebih lanjut, sistem ini dilengkapi fitur deteksi gangguan seperti penyumbatan *nozzle* spray atau indikasi kritis lingkungan. Peringatan dini serta jadwal pemeliharaan dikirim secara otomatis via *WhatsApp* langsung ke ponsel petani, memastikan informasi kritis tersampaikan tanpa penundaan. Implementasi ini memposisikan sistem tidak sekadar sebagai alat monitoring pasif, melainkan sebuah *decision support system* yang responsif dan berorientasi aksi.

Berdasarkan keseluruhan penelitian tersebut, dapat diketahui bahwa penelitian sebelumnya telah memberikan dasar penting dalam penerapan monitoring real-time, visualisasi data, sensor lingkungan, mikrokontroler ESP32, otomatisasi aktuator, dan integrasi database berbasis web. Namun, masing-masing penelitian masih memiliki fokus yang berbeda. Penelitian Ningsi lebih bersifat kajian mengenai smart farming dan pertanian presisi, penelitian Widagdo dkk. berfokus pada monitoring curah hujan, penelitian Kumara dkk. berfokus pada digitalisasi identitas dan pemeliharaan tanaman, penelitian Saputra dan Suryono berfokus pada otomatisasi irigasi berdasarkan kelembapan tanah, sedangkan penelitian Santoso dan Sultoni berfokus pada otomasi portal parkir berbasis RFID.

Dengan demikian, research gap penelitian ini terletak pada belum adanya integrasi berbagai fungsi tersebut secara khusus untuk pengendalian hama kutu pada tanaman bawang merah, yaitu penggabungan pemantauan beberapa parameter lingkungan, pengolahan data secara real-time, pengendalian penyemprotan otomatis, penyimpanan riwayat data, dashboard berbasis website, dan sistem

peringatan dini melalui WhatsApp API. Penelitian yang dilakukan diarahkan untuk mengisi kesenjangan tersebut melalui pengembangan Smart Sprayer IoT yang mengintegrasikan sensor lingkungan, ESP32, pompa sprayer, Laravel, MySQL, dan WhatsApp API dalam satu platform. Sistem yang dikembangkan memang ditujukan sebagai platform terpadu untuk monitoring dan penyemprotan otomatis pada bawang merah di Brebes.

2.1.2 Analisis Penelitian Terdahulu

Tabel 2.1.1
Analisis Penelitian Terdahulu

Penelitian	Fokus	Sensor/Komponen Utama	Otomatisasi	Platform/Media	Kekurangan / Research Gap
Ningsi (2025) [14]	Smart farming dan pertanian presisi	Data tanah dan iklim dari berbagai penelitian	Konseptual	Smart Farming, IoT, Precision Agriculture	Berupa kajian dan belum menghasilkan Smart Sprayer khusus bawang merah.
Widagdo dkk. (2018/2019) [15]	Monitoring curah hujan	Rain gauge, Raspberry Pi	Belum berfokus pada aktuator penyemprotan	Grafana, InfluxDB, Cloud VPS	Monitoring dan visualisasi curah hujan belum terintegrasi dengan penyemprotan otomatis.
Kumara dkk. (2025) [16]	Monitoring dan pemeliharaan pohon	Barcode, NFC, Dashboard	Terbatas pada digitalisasi pengelola	Dashboard Smart Farm	Belum menggunakan parameter lingkungan dan

			an tanaman		aktuator penyempr otan otomatis.
Saputra & Suryono (2024) [17]	Irigasi otomatis	Soil Moisture, ESP32, pompa	Otomatis berdasark an threshold	ESP32	Fokus pada irigasi tanaman jagung, bukan pengendal ian hama dan penyempr otan pestisida.
Santoso & Sultoni (2025) [18]	Portal parkir otomatis	RFID, ESP32, sensor kendaraan, portal	Otomatis	IoT dan database web	Bukan bidang pertanian dan tidak mengguna kan parameter lingkunga n tanaman.

Sumber : dikelola peneliti, 2026

Berdasarkan tabel tersebut, posisi penelitian ini berada pada irisan antara monitoring IoT, otomatisasi aktuator, sistem berbasis web, dan smart farming. Penelitian sebelumnya telah membuktikan bahwa sensor, ESP32, platform monitoring, dan otomatisasi dapat diterapkan secara efektif pada berbagai bidang. Namun, penelitian ini menggabungkan komponen-komponen tersebut ke dalam sistem yang lebih terintegrasi untuk kebutuhan pengendalian hama kutu pada bawang merah.

Hal tersebut sejalan dengan tujuan penelitian yang tercantum dalam dokumen, yaitu membangun dashboard berbasis web untuk memvisualisasikan parameter

lingkungan tanaman bawang merah secara real-time sekaligus mengintegrasikan modul IoT dan WhatsApp API untuk mengotomatiskan penyemprotan serta mengirimkan notifikasi peringatan dini kepada petani.

Kebaruan penelitian ini terletak pada integrasi beberapa parameter lingkungan, yaitu suhu, kelembapan udara, kelembapan tanah, dan curah hujan, dengan mikrokontroler ESP32, aktuator penyemprotan, dashboard berbasis website, database MySQL, serta WhatsApp API sebagai sistem peringatan dini. Integrasi tersebut dirancang dalam satu platform untuk mendukung monitoring kondisi lahan dan pengendalian penyemprotan secara otomatis maupun manual pada tanaman bawang merah dalam upaya pengendalian hama kutu.

2.2 Landasan Teori

2.2.1 Tinjauan Umum Sistem Monitoring Pertanian

Modernisasi pertanian saat ini mengarah pada adopsi sistem monitoring dan kontrol otomatis guna meminimalkan risiko gagal panen pada komoditas bawang merah akibat buruknya manajemen pengairan. Sebagai instrumen berbasis teknologi informasi, sistem pemantauan bertugas menghimpun dan mengolah data tanah secara berkelanjutan demi mendukung keputusan budidaya yang lebih akurat. Dalam ekosistem *IoT*, konsep monitoring telah berevolusi dari pencatatan data pasif menjadi arsitektur digital *real-time* yang mempermudah deteksi dini terhadap anomali tanaman maupun potensi serangan hama[19].

Penerapan platform monitoring berbasis *web* bertujuan meminimalkan asimetri informasi yang kerap terjadi pada pencatatan konvensional. Melalui

integrasi sensor suhu, kelembapan udara, kelembapan tanah, serta curah hujan, data mentah diolah di *server* dan disajikan dalam *dashboard* interaktif. Kelebihan utama sistem ini terletak pada kemampuannya mendeteksi fluktuasi parameter yang memicu siklus hidup hama bawang merah. Dengan demikian, petani memiliki fleksibilitas tinggi untuk memantau dan mengontrol kondisi lahan dari jarak jauh tanpa kendala geografis, menjaga stabilitas panen sekaligus menghemat input pertanian.

2.2.2 Konsep Integrasi Sistem dan API

Alur pengembangan dimulai dari identifikasi masalah sistem eksisting, diikuti analisis spesifikasi kebutuhan fungsional. Langkah selanjutnya meliputi perancangan *hardware*, basis data, serta *API* sebagai jembatan penghubung antara node sensor dan *web*. Aplikasi *web* terhubung dengan perangkat ESP32 via *RESTful API* untuk memfasilitasi transmisi data sensor kelembapan tanah, suhu, dan hujan secara *real-time*. Informasi ini diolah untuk divisualisasikan pada *dashboard* pemantauan hama bawang merah. Sebagai sistem peringatan dini (*early warning system*), *WhatsApp API* diintegrasikan guna mengirim notifikasi otomatis jika terdeteksi nilai ekstrem atau malfungsi pompa. Integrasi ini mempercepat deteksi anomali serta memperpendek waktu respons saat menghadapi situasi kritis di lahan[20].

2.2.3 Framework Laravel

Laravel merupakan *framework PHP open-source* dengan pola arsitektur *Model-View-Controller* (MVC) yang dirancang untuk membangun aplikasi *web* yang modular, aman, dan *scalable*. Dalam arsitektur sistem ini, *Laravel* bertindak

sebagai backend engine utama yang memproses data dari perangkat *IoT*. Model bertanggung jawab mengelola struktur data sensor (kelembapan tanah, suhu, status hujan) dan *log* penyemprotan di *database*. View berperan menyajikan antarmuka *dashboard* interaktif untuk visualisasi data *real-time* maupun grafik historis. Controller bertugas menangani request *RESTful API* dari ESP32, mengevaluasi logika otomatisasi pompa, dan memicu *WhatsApp API* jika terdeteksi kondisi kritis atau malfungsi alat. Dengan demikian, *Laravel* menjadi motor utama pemrosesan data dan kendali logika pada sistem pemantauan ini[21].

2.2.4 Landasan Teknologi Website dan Internet

a. Internet sebagai dasar sistem

Internet adalah jaringan komunikasi global yang memungkinkan pertukaran data antar perangkat melalui protocol.

b. Peran *website* dalam sistem IoT pertanian

Dalam sistem pemantauan pertanian berbasis IoT, *website* bertindak sebagai media utama yang menampilkan informasi mengenai kondisi lahan bawang merah secara digital, interaktif, dan secara langsung terhadap pengguna.

c. Fungsi *website* sebagai visualisasi data

Website adalah kumpulan halaman yang bisa diakses melalui peramban dan digunakan sebagai cara untuk menampilkan data dari sensor di lapangan, seperti kelembapan tanah, suhu udara, curah hujan, pengendalian hama, serta status penyemprotan yang dilakukan secara otomatis.

d. Arsitektur sistem (*client-server*)

Sistem menggunakan *arsitektur client-server*, dengan alur kerja:

1. Data sensor dikirim dari lahan ke server
2. Data diproses dan disimpan ke database
3. *System* menganalisis kondisi lahan (normal, waspada, kritis) terutama terkait potensi hama bawang merah.

2.2.5 Tailwind CSS

Pengembangan antarmuka pengguna (*UI*) dilakukan menggunakan *Tailwind CSS*, sebuah *framework CSS* modern berbasis utility-first untuk menghasilkan tata letak yang responsif dan efisien[22]. Berbeda dengan *framework UI* konvensional yang menyertakan komponen bawaan kaku, *Tailwind* menyediakan kelas utilitas tingkat rendah yang disematkan langsung pada elemen HTML. Metode ini meminimalkan *overhead* file *CSS* eksternal dan memberikan fleksibilitas penuh dalam penyesuaian visual. Di sisi fungsionalitas frontend, *TypeScript* diterapkan untuk menjamin *type safety* dan keandalan kode, meminimalkan error saat *runtime*. Sinergi antara fleksibilitas visual *Tailwind CSS* dan kekokohan struktur *TypeScript* menghasilkan komponen *web* yang adaptif, mudah dipelihara, dan tangguh untuk pemantauan data pertanian.

2.2.6 Basic Data MySQL

MySQL digunakan sebagai *relational database management system (RDBMS)* untuk menyimpan dan mengolah data sistem secara dinamis [20]. *Database* ini bertindak sebagai media penyimpanan relasional untuk merekam parameter lingkungan dari ESP32 secara terstruktur, termasuk tabel

sensor_readings untuk riwayat suhu/kelembapan, tabel devices untuk koordinasi status pompa, dan tabel *spray_logs* sebagai *audit trail* aktivitas penyemprotan. Keunggulan *MySQL* terletak pada performa *query* yang andal untuk *insert* data berkala maupun *select* data historis berukuran besar. Sinkronisasi data di *MySQL* memastikan konsistensi informasi yang digunakan sistem untuk memicu tindakan otomatis maupun manual. Pada tahap pengembangan, integritas data yang masuk ke *MySQL* dikawal melalui validasi tipe data menggunakan *TypeScript* di backend, sehingga menjamin konsistensi data yang berkelanjutan.

2.2.7 Teori Metode Waterfall

Dalam siklus rekayasa perangkat lunak, pemilihan metodologi yang tepat merupakan kunci keberhasilan proyek. Dibandingkan dengan model *Prototyping* yang cenderung fleksibel namun rentan terhadap *scope creep* dan pembengkakan biaya akibat perubahan kebutuhan yang berulang, model *Waterfall* menawarkan pendekatan yang linear dan sistematis. Model *Waterfall* dipilih dalam pengembangan sistem *Smart Sprayer* ini karena fasenya yang *sekuensial* dan terstruktur dengan baik, sehingga meminimalkan risiko manajemen proyek yang buruk serta menjamin stabilitas desain antarmuka sejak tahap awal[23].

a. *Requirements Analysis* (Analisis Kebutuhan)

Analisis Kebutuhan (*Requirements Analysis*): Tahap awal ini berfokus pada perincian kebutuhan sistem, seperti penetapan ambang batas sensor tanah untuk pemicu pompa, spesifikasi perangkat keras *IoT*, serta penentuan penerima pesan WhatsApp. Dokumentasi formal pada fase ini penting untuk mencegah

perubahan spesifikasi di tengah pengembangan yang dapat mengacaukan integrasi *hardware-software*.

b. *System Design* (Perancangan Sistem)

Desain Sistem (System Design): Desain sistem disusun sebagai *blueprint* teknis, meliputi perancangan skema rangkaian elektronik ESP32, pemodelan skema *database* relational di *MySQL*, serta *mockup* antarmuka *administrator* menggunakan *Tailwind CSS*.

c. *Implementation* (Implementasi/Coding)

Implementasi (Implementation/Coding): Merupakan fase pengodean berdasarkan *blueprint* yang telah disiapkan sebelumnya menggunakan *text editor Visual Studio Code*.

d. *Integration & Testing* (Integrasi dan Pengujian)

Integrasi & Pengujian (Integration & Testing): Modul *hardware* (sensor dan aktuator) dikoneksikan dengan sistem *software web*. Pengujian integrasi dilakukan secara komprehensif untuk memverifikasi akurasi pembacaan sensor, kelancaran grafik *real-time*, serta latensi pengiriman pesan notifikasi.

e. *Operation & Maintenance* (Operasi dan Pemeliharaan)

Operasi & Pemeliharaan (Operation & Maintenance): Setelah sistem terpasang stabil di lahan persawahan, pemeliharaan berkala dilakukan untuk mengidentifikasi *bug* aplikasi, membersihkan sensor dari residu pestisida, serta mengatasi penurunan performa *hardware* akibat pengaruh cuaca eksternal.

2.2.8 Unified Modeling Language (UML)

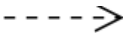
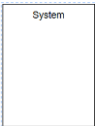
Perkembangan teknologi website sudah mengubah banyak hal dalam kehidupan, mulai dari cara informasi menyebar hingga meningkatkan efisiensi di berbagai sektor industri seperti pertanian. Dalam membuat situs web, memilih cara yang benar sangat penting untuk mengurangi kemungkinan masalah terjadi.

a) Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan interaksi fungsional antara aktor (pengguna) dengan sistem informasi yang dirancang[24]. Pemodelan ini merepresentasikan kelakuan (behavior) sistem secara eksternal, memetakan hak akses serta fungsi-fungsi utama yang tersedia dalam aplikasi. Notasi dan simbol yang digunakan dalam pemodelan *Use Case Diagram* dijabarkan pada Tabel 2.1.

Tabel 2. 1 Use Case Diagram

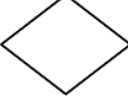
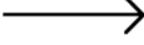
No	Gambar	Nama	Keterangan
1.		Actor	Merepresentasikan pihak luar yang berinteraksi dengan system, seperti petani atau admin. Actor dapat berupa manusia atau system lain.
2.		Use Case	Menggambarkan fungsi atau layanan yang disediakan oleh system terhadap actor, seperti monitoring sensor, control penyemprotan, dan melihat laporan.

No	Gambar	Nama	Keterangan
3.		Association	Menunjukkan hubungan komunikasi antara actor dengan use case dalam system.
4.		Include	Menunjukkan bahwa suatu use case selalu menyertakan use case lain sebagai bagian dari proses secara eksplisit.
5.		Extend	Menunjukkan hubungan perluasan dari use case lain yang hanya terjadi pada kondisi tertentu.
6.		generalization	Menunjukkan hubungan pewarisan antara actor atau use case, dimana elemen anak mewarisi sifat dari induknya.
7.		System Boundary	Menunjukkan Batasan system yang sedang dibangun dan memuat seluruh use case di dalamnya.
8.		Use Case Description	Menjelaskan detail aktivitas atau fungsi yang dilakukan dalam system dan memberikan hasil bagi actor.

b) Activity Diagram

Activity Diagram menggambarkan aliran kerja (*workflow*) atau rangkaian aktivitas operasional dalam suatu sistem dari awal hingga akhir. *Diagram* ini memodelkan keputusan-keputusan logis dan aksi paralel yang terjadi selama proses berjalan[25]. Simbol serta notasi penyusun *Activity Diagram* beserta fungsinya dirangkum dalam Tabel 2.2.

Tabel 2. 2 Activity Diagram

No	Gambar	Nama	Keterangan
1.		Initial Node	Menunjukkan titik awal dari suatu proses atau aktivitas dalam system.
2.		Final Node	Menunjukkan titik akhir dari suatu proses system.
3.		Activity	Menunjukkan aktivitas atau proses yang dilakukan dalam system.
4.		Decision	Digunakan untuk percabangan alur berdasarkan kondisi tertentu (misalnya: kelembapan < 40%).
5.		Control Flow	Menunjukkan arah aliran proses dari satu aktivitas ke aktivitas lainnya.
6.		Fork	Menunjukkan proses yang berjalan secara

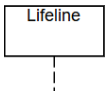
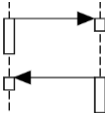
No	Gambar	Nama	Keterangan
			paralel (satu alur menjadi beberapa).
7.		Join	Menggabungkan beberapa proses paralel menjadi satu alur Kembali.
8.		Swimlane	Digunakan untuk membagi aktivitas berdasarkan pelaku atau bagian system (misal:petani,system,Io T).

c) Sequence Diagram

Sequence Diagram memodelkan interaksi antar objek berdasarkan urutan waktu (*chronological order*) untuk merealisasikan skenario *use case*. *Diagram* ini merinci lifespan objek serta pesan (*messages*) yang dikirim dan diterima antar objek di dalam sistem[26]. Daftar simbol dan notasi *Sequence Diagram* ditunjukkan pada Tabel 2.3.

Tabel 2. 3 Sequence Diagram

No	Gambar	Nama	Keterangan
1.		Actor	Pihak yang berinteraksi dengan system, seperti petani.
2.		Object	Komponen dalam system seperti website, server,

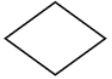
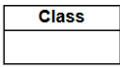
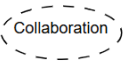
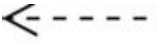
No	Gambar	Nama	Keterangan
			database, dan perangkat IoT.
3.		Lifeline	Menunjukkan keberadaan objek selama proses berlangsung.
4.		Message	Pesan atau perintah yang dikirim dari satu objek ke objek lain.
5.		Return Message	Pesan balasan yang dikirim Kembali oleh objek penerima.
6.		Activation	Menunjukkan periode objek sedang menjalankan proses.
7.		Destroy Object	Menunjukkan objek dihentikan atau tidak aktif lagi dalam system.
8.		Time	Menunjukkan urutan waktu dari atas ke bawah dalam diagram.

d) Class Diagram

Class Diagram merepresentasikan struktur statis sistem dengan memetakan kelas-kelas (*classes*) yang akan dibangun, atribut, metode (*methods*), serta hubungan asosiasi antar kelas tersebut. Pemodelan ini

memberikan cetak biru bagi penulisan kode berorientasi objek dalam program[27] Modeling class diagram using nlp in object-oriented designing. Simbol-simbol penulisan *Class Diagram* dapat dilihat pada Tabel 2.4.

Tabel 2. 4 Class Diagram

No	Simbol	Nama	Keterangan
1.		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendant</i>) mewarisi perilaku dan struktur dari objek induk (<i>ancestor</i>).
2.		<i>Nary Association</i>	Upaya ntuk menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi yang ditampilkan dalam sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		<i>Dependency</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan Dimana perubahan yang terjadi pada suatu elemen (<i>independent</i>) akan mempengaruhi elemen

No	Simbol	Nama	Keterangan
			yang bergantung padanya elemen yang tidak mandiri.
7.	—————	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

2.2.9 Pengujian Sistem (*Black Box Testing*)

phpMyAdmin merupakan perangkat lunak berbasis *web* open-source yang ditulis dalam *PHP*, berfungsi untuk mempermudah administrasi *database MySQL* dan *MariaDB* melalui antarmuka grafis (*GUI*). Aplikasi ini mendukung pengelolaan *database* secara efisien, mulai dari pembuatan tabel, pengeditan field, konfigurasi relasi antar tabel, pembuatan indeks, hingga manajemen *user privilege*[28].